Integrative machine learning approach for multi-class SCOP protein fold classification

Aik Choon Tan[1], David Gilbert[1] and Yves Deville[2]
[1]Bioinformatics Research Centre, Department of Computing Science,
University of Glasgow 17 Lilybank Gardens, G12 8QQ Glasgow, United Kingdom
{actan, drg}@brc.dcs.gla.ac.uk
[2] Department of Computing Science and Engineering,
Université catholique de Louvain Place Sainte Barbe 2,
B-1348 Louvain-la-Neuve, Belgium.
deville@info.ucl.ac.be

Abstract: Classification and prediction of protein structure has been a central research theme in structural bioinformatics. Due to the imbalanced distribution of proteins over multi SCOP classification, most discriminative machine learning suffers the well-known 'False Positives' problem when learning over these types of problems. We have devised eKISS, an ensemble machine learning specifically designed to increase the coverage of positive examples when learning under multi-class imbalanced data sets. We have applied eKISS to classify 25 SCOP folds and show that our learning system improved over classical learning methods.

1. Introduction

Learning the similarities (or differences) between protein structures is very important in understanding the relationship between protein sequence, structure and function, and for the analysis of possible evolutionary relationships. Several new initiatives (e.g. structural genomics) and improvement of the methods for structure determination will result in a rapid increase in the number of structures. These high-throughput structure determination projects will produce structural data on proteins for which very little is known about their biology. Thus sophisticated computational methods are needed to detect, search for and compare remote protein homology in the hope that knowledge can be transferred to the new unknown protein (e.g. inference about function).

Machine learning is one such approach that has been widely used in the development of automatic protein structure classification and prediction (Turcotte et al, 2001; Selbig and Argos, 1998; King et al., 1994). One of the aims of structural genomics is to enhance the understanding of the relationships between amino acid sequence and its corresponding protein fold. Hence, one of the advantages of using symbolic machine learning approaches for this purpose is to generate human understandable classifiers (rules) from some biological background knowledge that can explain the current protein folds in the Protein Data Bank (PDB).

The SCOP database (Lo Conte et al, 2002) is a comprehensive hierarchical human classification of known protein structures, according to their evolutionary and structural relationships. The SCOP database is divided into 4 hierarchical levels: Class, Fold, Superfamily and Family. For SCOP 1.61 (Sept 2002), the 44327 protein domains were classified into 701 folds, resulting in an average of 64 domains per fold. The number of domains per fold varies in SCOP, where some of the folds are highly populated (e.g. TIM barrels) while some of the folds only contain a few examples (e.g. the HSP40/DnaJ peptide-binding fold only contains one protein). Thus, in order to perform learning over the SCOP folds, the common one-against-others approach (two-class problem) will result in learning with an imbalanced data set. This imbalanced proportion of examples in each class contributes to the poor performance of standard machine learning techniques (e.g. decision trees). Existing machine learning approaches tend to produce a strong discriminatory classifier (high accuracy) with very low sensitivity (also called completeness) when learning on these types of problems.

The specific problem we would like to address here is learning from multi-class SCOP fold imbalanced data sets where the protein examples from one class heavily outnumber those from the other class (e.g. 1 to 5%). The goal of this work is to develop a learning system to classify multi-class problems in an imbalanced data situation. We have devised eKISS (**e**nsemble **K**nowledge for **I**mbalance **S**ample **S**ets), an ensemble learning method to tackle these types of problems. The objective of eKISS is to generate one-against-others classifiers which are capable of learning over multi-class examples under the skewed normal distribution of the training examples, as well as providing explanation to the user.

2. Machine Learning Background

For a supervised classification problem, a set of training data (positive and negative examples) in the form of {x, y | x $\in$ attributes, y $\in$ classes} is provided to the learner *L*. The learner's task is to induce a set of rules that can discriminate positive examples (E+) from negative ones (E-), and thus propose a classification for new instances. The common approach of treating multi-class learning is to transform the *K* classes into a set of two-class problems, which is also known as one-against-others method. This approach faces one serious pitfall when learning in multi class problems: when we transform the *K* classes into *K* two-class problems, the positive examples of a class C_1 will be under-represented compared to the large number of negative examples for class $C_2,\ldots,C_K$. The presence of large amount of negative examples in the training data poses several pitfalls for classical machine learning systems.

The major problem of applying discriminative classical machine learning techniques (e.g. decision trees, artificial neural networks) in this situation is they either generate a trivial rejector classifier, which classifies everything as a negative class (due to the negative examples being the majority class); or they overfit the positive examples by generating large decision trees or highly complex neural networks. Most discriminative learning approaches apply recursive partitioning of the instance space into regions labelled with the majority class in that region. Furthermore, the heuristic of stopping or pruning criteria for the partitioning procedure is constructed to avoid 'overfitting' the training examples which is solely based on the overall accuracy or the overall error rate of the classifier, which represents a weak measurement under the imbalanced data. This heuristic, known as Occam's razor in the machine learning literature, suggests that a learning algorithm should prefer "simpler" to more "complex" classifiers in order to avoid overfitting the training examples. Wolpert's "No-free-lunch" theorems pointed out that all such heuristics fail as often as they succeed in supervised learning problems (Wolpert, 2001). Hence, most classical machine learning methods suffer the above mentioned drawbacks and perform poorly under the two-class imbalanced data situation. This scenario is described as the "curse of imbalanced data" in machine learning terminology (Kubat et al, 1998).
To overcome the two-class imbalanced data set problem, some attempts have been proposed in the machine learning community which involve either (i) reducing the negative class by randomly removing the negative examples from the training set; or (ii) increasing the positive class by replicating the positive examples. Removing or increasing the training examples is not suitable in this research domain due to the multi-class nature of the training examples and the limited number of the real protein data.

Another approach for handling multi-class problems is to generate all the possible pairwise two-class classifiers between *K* classes from the training examples. This approach is known as all-versus-all method in which given *K* classes of training examples, the machine learning methods will generate two-class classifiers for all the $K(K-1)/2$ classifiers. The unseen proteins have to be classified by these classifiers; every classifier provides a vote for the class label, and the majority voted class will be the predicted class for the new proteins. In the ideal case, the correct class will get the maximum votes for all the class-paired classifiers. In our case, we observed that this approach does not perform well due to the votes of the correct class being

randomly distributed among other classes. Most classifiers always produced a trivial rejector which votes for a negative class. This problem is also observed by Ding and Dubchak (2001) where they described the votes for the most popular voted class decreasing gradually from maximum to minimum and simply returning the class with the highest vote. The other disadvantage of this approach is the large number of classifiers, which is very difficult and hard to analyse for the purpose of providing insights into understanding the protein sequence-structure relationships.

3. Methods

In this paper, we propose eKISS, an ensemble machine learning approach, which integrates the classifiers generated from the one-against-other and all-against-all approaches to improve the coverage of the positive protein examples under the multi-class imbalanced data. Ensemble machine learning can be loosely defined as a set of classifiers whose individual decisions are combined in some way to classify new examples (Dietterich, 2000). Several empirical studies have shown that the performance of ensemble machine learning approaches is better than that of single methods due to the drawbacks discussed in the background section as well as the existing "No-free-lunch" theorems in the individual learning algorithms (Tan and Gilbert, 2003; Bauer and Kohavi, 1999; Quinlan, 1996).

In our approach, we have applied the PART rule-based machine learning technique to generate the base classifiers for our ensemble learning system. PART (Frank and Witten, 1998) is a rule-induction algorithm that avoids global optimisation, and generates accurate and compact rule sets by combining the paradigms of "divide-and-conquer" (C4.5, Quinlan, 1993) and "separate-and-conquer" (RIPPER, Cohen, 1995). PART adopts the separate-and-conquer strategy in that it builds a rule, removes the covered instances, and continues constructing rules recursively by generating a partial decision tree from the remaining instances. The number of rules generated from PART is fewer and more compact compared to RIPPER and C4.5. We have performed a one-against-others procedure to generate K two-class classifiers and also an all-against-all approach to produce $K(K\text{-}1)/2$ classifiers. We then combined these $K + K(K\text{-}1)/2$ base classifiers to generate a new classifier per class, called the ensemble classifiers. For this protein fold classification problem, the ensemble contains $25 + (25 \times 24)/2 = 325$ base classifiers. Since PART is a rule-based learning system, each PART classifier contains a set of decision rules. To simplify the presentation, we assume that each base PART classifier contains k positive decision rules, denoted R_{i1}, R_{i2}, …, R_{ik} for the base classifier number i.

Classical machine learning methods generate a classifier by performing a heuristic search through the possible classification rules (true hypotheses) of the given instance space, trying to find rules that can "best" approximate the true classification of the instance space. Since the heuristics employed so far are not suitable for the multi-class imbalanced data sets, the classical machine methods suffer the "curse of learning in imbalanced data" and most of the time return a near optimal trivial rejector classifier.

The basic idea of eKISS is to consider any rule R_{ij} as a potential candidate rule for each of the new ensemble classifiers. The main assumption made in eKISS is that all the rules generated by the PART learning algorithm represent possible classification rules, hence enlarging the search space. The eKISS search strategy is to find all the rules that correctly classify the examples in the positive class, hence improving the coverage of the positive examples under the multi-class imbalanced data situation. We also believe these positive rules are useful for providing insights to the human expert in understanding the relationships between protein structure and sequence information compared to a trivial rejector classifier. Technically, a rule R_{ij} will be included in the new ensemble classifier of a given class if it correctly classifies the positive examples of that class. As a decision measure, we use the normalised confidence measurement, cf_norm = (TP-0.5)/(TP+FP(E+/E-)) as the cut-off point for rule selection. The rules of the new classifier for class C_i are all the rules that satisfy the cut-off point. The normalised confidence measurement has been applied by

Quinlan (1993) in evaluating the goodness of the decision rules derived from the decision trees. This measurement takes into account the ratio of the positive and negative examples and thus produces a much more sensitive measurement for computing the accuracy of the rules in an imbalanced data situation. Obviously, some (but not all) of the rules of the base classifier of a class will be in the new classifier of the same class, as well as rules from other base classifiers. In our system, cf_norm represents the tuning parameter for trade-off between the coverage of the positive examples (TP-rate) and the precision (positive predicted value). eKISS allows the user to select the classifier that best suits his/her classification purpose by alternating the cf_norm value. Furthermore, in order to assist the user in selecting his/her choice of classifiers, the system can automatically generate ROC (Receiver Operating Characteristic) curves that provide an initial visualisation tool to facilitate the selection process. Note that the ensemble approach of eKISS is slightly different from the traditional ones; instead of combining decisions from different base classifiers, we combine the rules of the classifiers to generate new classifiers.

The proposed method has been designed to increase the sensitivity (positive coverage) of the classifiers. One would then expect the method to have a reduced specificity (also called soundness). As will be shown in the results, this approach is useful when the ratio E+/E- is very low, and also when the initial classifiers yield little sensitivity. In that case, the loss of specificity is small compared to the increase of sensitivity, yielding more useful classifiers. Obviously, for some classes the base classifiers can be preferred to the new one.

4. Data set

The protein data set that we used in this study was from Ding and Dubchak (2001), which can be obtained from http://www.nersc.gov/~cding/protein. The original data used by Ding and Dubchak (2001) contains two different sets, a training set and a test set. The training set was extracted from the PDB_select sets (Hobohm and Sander, 1994) by filtering out all the proteins that have less than seven examples for each SCOP classification. All of the pair-wise protein sequence identities of this set are less than 35%. From this set, Ding and Dubchak compiled 313 proteins from 27 most populated SCOP folds that they referred to as N_{train} in their paper. The test set N_{test} in their paper was extracted from PDB_40D (Lo Conte et al., 2000) which contains 386 representatives of the same 27 SCOP folds with sequence similarity less than 35% (filtering out all the proteins with sequence identity more than 35% of PDB_40D and excluding the proteins in the N_{train}). The attributes used in the learning system are extracted from protein sequences according to the method described in Dubchak *et al* (1997) where a protein sequence is represented by a set of parameter vectors on various physico-chemical and structural properties of amino acids along the sequence. These properties are hydrophobicity, polarity, polarizability, predicted secondary structre, normalised van der Waals volume and the amino acid composition of the protein sequence. In this study, we combined all the protein sequence parameters resulting 125 physico-chemical and structural properties of amino acids as our learning attributes.

Before exploiting these data, we analysed both the training and test sets and found some interesting errors in both data sets, especially in the training set (N_{train}). The first error is the inconsistency of the data sets. Ding and Dubchak (2001) used the protein data from PDB_selects as the training set, at a time when the SCOP classification did not exist. Although Ding and Dubchak (2001) reclassified the training set according to the early SCOP database, we still believe the domain definition in SCOP was still not well defined. Their test set was extracted from the more recent SCOP database (SCOP 1.48, Dec 1999) for which the domain definitions are well defined and which clearly contains major changes compared to the early SCOP version which had been used to assign the training set. We found some protein examples in the training set which had not been assigned into domains at that time (due to the earlier domain definitions by SCOP) but were present in the

test set as different chopped domains. Probably this "dirty" data may have contributed to some poor performance of Ding and Dubchak's (2001) analysis. At the same time, this also shows that the domain definition has evolved in the SCOP database over time by careful manual assignment; an automatic and intelligent system may facilitate this protein fold classification process.

Therefore, we have extracted the data set by removing the error from both training and testing examples. We applied the protein fold classification according to the SCOP 1.61 (Nov 2002, Lo Conte et al, 2002) and Astral 1.61 (Chandonia et al 2002) with sequence identity less than 40% (Nov 2002), removing those fold class with less than 8 examples. After performing this cleaning stage, our protein fold data contains 582 examples distributed in 25 fold SCOP classes. We randomly divided the data into a training set (408 protein examples) and a test set (174 protein examples).

Standard measurements have been applied to evaluate the goodness of our classifiers compared to PART: true positive rate (also called positive coverage or sensitivity, TPR = TP/(TP+FN)), false positive rate (FPR = FP/(FP+TN) or (1 – specificity)), positive predicted value (PPV = TP/(TP+FP)) and F_1-measure ((2Sn x PPV)/(Sn + PPV)) (van Rijsbergen, 1979) which evaluates the trade off between sensitivity and positive predicted value.

5. Results and Discussion

We performed ten-fold cross-validation on the training data and evaluated the test set by comparing the performance of eKISS and PART. Table 1 summarises the performance on the training and test sets. From the results, eKISS outperforms PART on 20 classes based on the F_1-measure. The results show that eKISS increases the sensitivity and also the positive predictive accuracy compared to PART. Although our method increases the true positive rate (TPR), as a trade-off it also increases the false positive rate (FPR). Since the objective of this study is to improve the rule coverage when classifying protein folding classes, we permit the rule-set to cover some false positives as a consequence of improving the positive coverage of classical machine learning. However, the results show that the increase of TP-rate is higher than the corresponding increase of the FP-rate.

In order to verify the hypothesis that the set of rules from all the base classifier forms a useful search space for the generation of the new classifiers, we also used a set of random rules (obtained by applying PART on a randomly generated data set). The performances of the resulting new classifiers were clearly under the performance of eKISS.

In general, eKISS performs well in learning from a small set of positive examples compared to the negative examples because eKISS is capable of generating a softer boundary for the classifier. It thus avoids problems connected with the strong discriminative boundary generated by classical learning systems. One of the essential conditions for ensemble methods to perform better than any of its individual classifier members is the diversity of the base classifiers. Two classifiers are diverse if they perform different prediction errors on new instances. The advantage of having diverse base classifiers is illustrated as follows; let us assume an ensemble of three base classifiers $\{h_1, h_2, h_3\}$ and a new instance x. If the base classifiers are identical (i.e. not diverse), then when the prediction of $h_1(x)$ for its corresponding class label y is wrong, $h_2(x)$ and $h_3(x)$ will also wrongly predict the class label y. Thus, the ensemble h*(x) of these base classifiers will not improve the prediction of class y for x. However, if the errors made by the base classifiers are uncorrelated and when the prediction of $h_1(x)$ is wrong, then the prediction of $h_2(x)$ and $h_3(x)$ might be correct. Therefore the ensemble h*(x) which obtained the final prediction from collecting the majority vote of its base classifiers will correctly classify x (Dietterich, 2000). We believe that the base classifiers of eKISS are made diverse by combining the one-against-others and the all-against-all PART classifiers. Re-selecting the appropriate rules from these base classifiers creates the diversity of the ensemble and hence improves the positive coverage of eKISS.

Another interesting finding from this experiment is that the rule sets generated from eKISS are much smaller than those of the original PART system. We would have expected eKISS rule-sets to contain more rules compared to PART due to "collecting" additional rules from other classifiers, but it turns out they have increased sensitivity. We believe that these rule-sets are useful for classifying protein folds and thus can assist wet experimental biologists in understanding the co-relationships between amino acid physico-chemical properties and functions.

Table 1: Performance evaluation of eKISS (cf_norm = 0.69) and PART over training and testing sets.

SCOP (SCOP id)	**Method**	**Training Set**				**Test Set**			
		TPR	**FPR**	**PPV**	**F_1-measure**	**TPR**	**FPR**	**PPV**	**F_1-measure**
Globin-like (a.1)	eKISS	0.611	0.255	0.076	0.131	0.833	0.399	0.069	0.128
	PART	0	0.038	0	Undefined	0.167	0.048	0.111	0.133
Cytochrome c (a.3)	eKISS	0.500	0.246	0.056	0.100	1.000	0.491	0.057	0.108
	PART	0.100	0.023	0.033	0.050	0	0.006	0	Undefined
DNA-binding 3-helical bundle (a.4)	eKISS	0.944	0.468	0.133	0.227	1.000	0.485	0.101	0.184
	PART	0.050	0.056	0.050	0.050	0	0.066	0	Undefined
4-helical up-and-down bundle (a.24)	eKISS	0.333	0.091	0.083	0.131	0.750	0.259	0.064	0.118
	PART	0	0.010	0	Undefined	0	0.029	0	Undefined
4-helical cytokines (a.26)	eKISS	0.500	0.322	0.061	0.105	1.000	0.420	0.066	0.123
	PART	0	0.021	0	Undefined	0	0.029	0	Undefined
EF hand-like (a.39)	eKISS	0.333	0.166	0.060	0.095	0	0	0	Undefined
	PART	0	0.015	0	Undefined	0.100	0.030	0.100	0.125
Immunoglobulin-like β-sandwich (b.1)	eKISS	0.790	0.443	0.155	0.248	1.000	0.692	0.120	0.214
	PART	0.200	0.094	0.039	0.065	0	0.084	0	Undefined
Cupredoxin-like (b.6)	eKISS	0.520	0.367	0.066	0.111	0.833	0.548	0.052	0.097
	PART	0.033	0.019	0.100	0.050	0	0.060	0	Undefined
Viral coat & capsid proteins (b.10)	eKISS	1.000	0.713	0.109	0.193	1.000	0.813	0.056	0.106
	PART	0.033	0.049	0.100	0.05	0	0.047	0	Undefined
Concanavalin A-like lectins/glucanases (b.29)	eKISS	0.400	0.175	0.038	0.068	0.667	0.275	0.041	0.077
	PART	0.100	0.007	0.100	0.100	0	0.006	0	Undefined
SH3-like barrel (b.34)	eKISS	0.667	0.203	0.076	0.137	0.750	0.253	0.065	0.12
	PART	0.100	0.013	0.100	0.100	0	0.018	0	Undefined
OB-fold (b.40)	eKISS	0.148	0.160	0.016	0.027	0.667	0.376	0.088	0.156
	PART	0.050	0.058	0.050	0.050	0.067	0.082	0.071	0.069
β-Trefoil (b.42)	eKISS	0.400	0.068	0.075	0.124	1.000	0.265	0.082	0.151
	PART	0	0.020	0	Undefined	0	0.018	0	Undefined
Lipocalins (b.60)	eKISS	0	0.034	0	Undefined	1.000	0.265	0.082	0.151
	PART	0.050	0.016	0.050	0.050	0	0.024	0	Undefined
TIM α/β-barrel (c.1)	eKISS	0.933	0.541	0.178	0.295	1.000	0.768	0.138	0.242
	PART	0.083	0.159	0.056	0.067	0	0.124	0	Undefined
NAD(P)-binding Rossmann-fold (c.2)	eKISS	0.357	0.173	0.027	0.050	0	0	0	Undefined
	PART	0	0.025	0	Undefined	0.333	0.018	0.400	0.364
FAD/NAD(P)-binding domain (c.3)	eKISS	0.313	0.220	0.021	0.040	0.500	0.310	0.055	0.098
	PART	0.157	0.005	0.400	0.195	0	0.024	0	Undefined
Flavodoxin-like (c.23)	eKISS	0.670	0.477	0.091	0.154	0	0	0	Undefined
	PART	0.225	0.079	0.117	0.145	0	0.071	0	Undefined
P-loop containing nucleotide (c.37)	eKISS	0.571	0.362	0.045	0.082	1.000	0.692	0.041	0.079
	PART	0	0.017	0	Undefined	0	0.047	0	Undefined
Thioredoxin-fold (c.47)	eKISS	0	0	0	Undefined	1.000	0.265	0.082	0.151
	PART	0	0.008	0	Undefined	0.111	0.012	0.333	0.111
Ribonuclease-H-like motif (c.55)	eKISS	0	0.053	0	Undefined	0	0	0	Undefined
	PART	0.033	0.015	0.100	0.050	0	0.029	0	Undefined
α/β-Hydrolases (c.69)	eKISS	0.920	0.271	0.142	0.239	0.750	0.529	0.032	0.062
	PART	0.050	0.020	0.025	0.033	0	0.018	0	Undefined
β-grasp (ubiquitin-like) (d.15)	eKISS	0	0.080	0	Undefined	0.500	0.276	0.041	0.075
	PART	0	0.007	0	Undefined	0	0.006	0	Undefined
Ferrodoxin-like (d.58)	eKISS	0.313	0.088	0.067	0.108	0.600	0.317	0.103	0.176
	PART	0.025	0.128	0.014	0.018	0	0.059	0	Undefined
Knottins (small inhibitors, toxins, lectins) (g.3)	eKISS	1.000	0.496	0.166	0.281	1.000	0.640	0.112	0.202
	PART	0.100	0.083	0.100	0.100	0	0.053	0	Undefined

The disadvantage of using overall accuracy and error rate as the heuristic for stopping or pruning criteria is due to the large value of the denominator in both measurements. Assume that the training examples contain 990 negative examples and 10 positive examples. Then a trivial rejector classifier that classifies all the training examples as negative class will have an overall accuracy of

(990/1000)×100% = 99% and error rate of only 1%, which represents a high accurate classifier. This trivial rejector classifier tends to outperform all non-trivial classifiers but it is a meaningless classifier which is unable to discriminate between instances in both classes. This example also illustrates that the overall accuracy and error rate is not a sensible measure of the effectiveness or usefulness of a classifier under the imbalanced data set. The information retrieval (IR) community has a long history in classifying documents under imbalanced sample sets. F_1-measure (van Rijsbergen, 1979) is the popular measurement that the IR community applies to evaluate the trade-off between TP-rate and FP-rate. We believe this measurement is a sensible evaluation method in this problem as our system constantly outperforms PART.

6. Conclusions

We have proposed eKISS, an ensemble method that has been specifically designed to increase the sensitivity (positive coverage) of the classifiers without losing much of its corresponding specificity when learning over multi-class imbalanced data sets where protein examples from one class heavily outnumber examples from the other class. We have applied this approach to classification of 25 SCOP protein folds and our preliminary results show that this approach is useful when the ratio E+/E- is very low, and also when the initial classifiers yield little sensitivity. In that case, the loss of specificity is small compared to the increase of sensitivity, yielding more useful classifiers. The rules generated by eKISS are shorter and may provide hints to the understanding of amino acid physico-chemical properties and it's constituted fold.

Acknowledgements

We would like to thank Gilleain Torrance and Ali Al-Shahib for their useful comments and proof reading of this manuscript. AC Tan was funded by a University of Glasgow studentship.

References

Bauer, E. and Kohavi, R. (1999). Machine Learning, 36: 105-142.

Chandonia, J.-M., et al. (2002). Nuclei Acids Research, 30: 260-263.

Cohen, W.W. (1995). In Proceedings of the 12th ICML, p.115-123. Morgan Kaufmann

Dietterich, T.G. (2000). LNCS 1857, p.1-15.

Ding, C.H.Q. and Dubchak, I. (2001). Bioinformatics, 17: 349-358.

Dubchak, I., et al. (1997). In Proceedings of the 5th ISMB, p. 104-107. AAAI.

Frank, E. and Witten, I.H. (1998). In Proceedings of the 15th ICML, p.144-151.

Hobohm, U., et al. (1992). Protein Science, 1: 409-417.

Kubat, M. et al (1998), Machine Learning, 30: 195-215.

King, R.D., Clark, D.A., Shirazi, J. and Sternberg, M.J.E. (1994). Protein Engineering, 7: 1295-1303.

Lo Conte, L., et al. (2000). Nuclei Acids Research, 28: 257-259.

Lo Conte, L., et al (2002). Nuclei Acids Research, 30: 264-267.

Quinlan, J.R. (1993). C4.5: Programs for Machine Learning, San Mateo, CA: Morgan Kaufmann.

Quinlan, J.R. (1996). In Proceedings of the 13th National Conference on AI. P.725-730.

Selbig, J. and Argos, P. (1998). PROTEINS: Structure, Function, and Genetics. 31: 172-185.

Tan, A.C. and Gilbert, D. (2003). In Proceedings of the 1st Asia Pacific Bioinformatics Conference, p.219-222.

Turcotte, M., et al. (2001). J. Mol. Biol., 306: 591-605.

van Rijsbergen, C.J. (1979). Information Retrieval, 2nd Ed.. Butterworths.

Wolpert, D. H. (2001) In Proceedings of the Sixth Online World Conference on Soft Computing in Industrial Applications.